REMARKS

Reconsideration and allowance of the present application are respectfully requested. Claims 1-39 remain pending in the application. By this Amendment, claims 1, 2, 13, 14, 25 and 26 are amended.

In numbered paragraph 4, on page 3 of the Office Action, independent claims 1, 13 and 25, along with various dependent claims, are rejected as being unpatentable over U.S. Patent No. 5,758,051 (Moreno et al.), in view of U.S. Patent No. 5,787,287 (Bharadwaj) and U.S. Patent No. 6,077,313 (Ruf). In numbered paragraph 5, on page 11 of the Office Action, dependent claims 3, 4, 15, 16, 27 and 28 are rejected as being unpatentable over the Moreno et al. patent, in view of the Bharadwaj patent and the Ruf patent, and further in view of U.S. Patent No. 6,016,398 ('398 Radigan). In numbered paragraph 6, on page 13 of the Office Action, dependent claims 6, 18 and 30 are rejected as being unpatentable over the Moreno et al. patent, in view of the Bharadwaj patent and the Ruf patent, and further in view of U.S. Patent No. 6,151,704 ('704 Radigan). These rejections are respectfully traversed.

Applicant has disclosed exemplary embodiments for modifying serial dependencies in a procedure. Included is an exemplary second memory operation which is moved to a new position in a graph representation that is closer to a first memory operation. The step of moving the second memory operation includes (i) removing one or more of serial dependencies in initial set of serial dependencies that is associated with the second memory operation and (ii) creating a new serial dependency between the first memory operation and the second memory operation. As exemplified in FIG. 10A, it is possible to remove a dependency edge from 1006 to

1008 and replace it with a new dependency edge from 1002 to 1008. A dashed arrow from 1002 to 1008 represents this new dependency edge. Upon removal of the edge, or dependency, from 1006 to 1008, a compiler 58 can move load 1008 out of the repetitive loop, as shown in FIG. 10B (e.g., specification at page 20, lines 25-33). Such a movement of memory operation as shown in FIG. 10B can provide enhanced and efficient machine code compared to the prior depiction of FIG. 10A because load 1008 is executed only once in FIG. 10B rather than on a repetitive basis as in FIG. 10A (e.g., specification at page 21, lines 1-3). The exemplary movement to a new position with the change of dependencies can thus provide enhanced and efficient machine code.

The foregoing features are broadly encompassed by claim 1 which recites, among other features, a method for modifying serial dependencies in a procedure, including moving a second memory operation to a new position in a graph representation that is closer to a first memory operation, wherein the moving step includes (i) removing one or more of serial dependencies in an initial set of serial dependencies that is associated with the second memory operation and (ii) creating a new serial dependency between the first memory operation and the second memory operation.

The documents relied upon by the Examiner, regardless of whether they are considered alone or in the combination discussed by the Examiner, fail to teach or suggest features of claim 1. The Examiner admits at page 5 of the Office Action that "Moreno does not expressly disclose the limitations wherein:...the moving step includes (i) removing one or more of serial dependencies in an initial set of serial dependencies that is associated with the second memory operation and (ii) creating a new serial dependency between the first memory operation and the second memory operation."

The Bharadwaj patent does not cure the deficiencies of the Moreno et al. patent. The Bharadwaj patent merely discloses that dependency path vectors can be used to "determine if particular code motion or code reordering may be performed without altering the meaning of the source code" (col. 6, lines 41-43). Instead of removing or creating a serial dependency as claimed, the Bharadwaj patent depends on "inserting extra register copy operations" to preserve the meaning of the source code upon code motion (col. 6, lines 43-45). The Moreno et al. and Bharadwaj patents therefore fail to teach or suggest Applicant's claim 1 combination of features, such that claim 1 is allowable.

Further, the Moreno et al. and Bharadwaj patents do not provide the suggestion to remove or create a serial dependency upon code motion. The insertion of extra register copy operations as taught by the Bharadwaj patent inserts an extra operation between operations, and teaches away from reconfiguring serial dependencies upon code motion. Accordingly, one of ordinary skill in the art at the time of the invention would not have been motivated to combine the Moreno et al. patent and the Bharadwaj patent to derive the claimed (i) removing one or more of serial dependencies in an initial set of serial dependencies that is associated with the second memory operation and (ii) creating a new serial dependency between the first memory operation and the second memory operation.

The Ruf patent, the '398 Radigan patent, and the '704 Radigan patent do not cure the deficiencies of the Moreno et al. patent, even when considered in the combination relied upon by the Examiner. The Ruf patent was cited for its disclosure of a dependence analysis module 314 and a partitioning algorithm module 316 (col. 9, line 66 to col. 10, line 21); the '398 Radigan patent was cited for its disclosure of an intermediate language representation (e.g., col. 6, lines 56-61); and the '704 Radigan patent was cited for its disclosure of optimizing a loop in a computer program (e.g., col. 6, lines 3-11). The Ruf patent, the '398 Radigan patent, and the '704 Radigan patent do not teach or suggest at least the moving step, including removing one or more of serial dependencies in an initial set of serial dependencies that is associated with the second memory operation and creating a new serial dependency between the first memory operation and the second memory operation.

Even if the references could have been combined in the manner asserted by the Examiner, the combination would not have resulted in a method for modifying serial dependencies in a procedure, including moving a second memory operation to a new position in a graph representation that is closer to a first memory operation, wherein the moving step includes (i) removing one or more of serial dependencies in an initial set of serial dependencies that is associated with the second memory operation and (ii) creating a new serial dependency between the first memory operation and the second memory operation. For example, combining of features from the Moreno et al. patent, the Bharadwaj patent and the Ruf patent, in the manner asserted by the Examiner would not have resulted in a method for modifying serial dependencies in a procedure, including moving a second memory operation to a new position in a graph representation that is closer to a first memory operation, wherein the moving step includes (i) removing one or more of serial dependencies in an initial set of serial dependencies that is associated with the second memory operation and (ii) creating a new serial dependency between the first memory operation and the second memory operation.

For the foregoing reasons, Applicant's claim 1 is allowable. Independent claims 13 and 25 recite features similar to those discussed above, and are therefore also allowable. The remaining dependent claims recite additional advantageous features which further distinguish over the documents relied upon by the Examiner. As such, the present application is in condition for allowance.

All objections and rejections raised in the Office Action having been addressed, it is respectfully submitted that the application is in condition for allowance and a Notice of Allowance is respectfully solicited.

Respectfully submitted,

BURNS, DOANE, SWECKER & MATHIS, L.L.P.

for

Date: April 29, 2005

By: Reg. No. 48,360
Patrick C. Keane
Registration No. 32,858

P.O. Box 1404
Alexandria, Virginia 22313-1404
(703) 836-6620